Manual Inspection of ARCOS Opioid Dataset

Michael Rogove | Megana Lakshmi Padmanabhan | Kevin Hederman

Purpose of Notebook:

- Demonstrate use of PySpark to handle large files
  - With right configuration, can process sophisticated queries against 75GB file in minutes or less.
- Uncover more insights and further questions for research and investigation.
  - Parameterized where possible.

Recommended configuration:

- GCP Dataproc, standard N nodes (not high-avail)
  - 1000GB disk Master
  - 4 500GB workers.
- Enable Jupyter and Anaconda.
- Enable API access among project components.

In [2]:
```
import os
import sys
import pyspark.sql.functions
```

Preparing the data

Assumes unzipped data in GCP bucket. Steps to achieve this outlined in presentation.

Ingest dictionary and source file

(beginning with the data dictionary as reference)

In [34]:
```
datadict = spark.read.option("sep", ",").option("header","true").csv("gs://119-f19-opioidbucket/dat
a_dictionary.csv")
datadict.show(50,False)
```

ColumnName	Description
REPORTER_DEA_NO	Unique id of entity reporting shipments to DEA. Reporters must have unique id for each facility, so some repoerters have multiple ids.
REPORTER_BUS_ACT	Type of business the reporter does, typically distributors or manufacturers.
REPORTER_NAME	Name of entity reporting shipments to the DEA.
REPORTER_ADDL_CO_INFO	Additional company information for entity reporting shipments to the DEA.
REPORTER_ADDRESS1	Address of entity reporting shipments to the DEA.
REPORTER_ADDRESS2	Additional address field for entity reporting shipments to the DEA.
REPORTER_CITY	City of entity reporting shipments to the DEA.
REPORTER_STATE	State of entity reporting shipments to the DEA.
REPORTER_ZIP	Zip code of entity reporting shipments to the DEA.
REPORTER_COUNTY	County of entity reporting shipments to the DEA.
BUYER_DEA_NO	Unique id of entity receiving shipments from reporter.
BUYER_BUS_ACT	Type of business the reporter does. Our data set limits to retail pharmacies, chain pharmacies and types of practitioners, though full data set includes more including mail order pharmacies, hospitals and distributors, among others.
BUYER_NAME	Name of entity receiving shipments from reporter.
BUYER_ADDL_CO_INFO	Additional company information for entity receiving shipments from reporter.
BUYER_ADDRESS1	Address of entity receiving shipments from reporter.
BUYER_ADDRESS2	Additional address field for entity receiving shipments from reporter.
BUYER_CITY	City of entity receiving shipments from reporter.
BUYER_STATE	State of entity receiving shipments from reporter.
BUYER_ZIP	Zip code of entity receiving shipments from reporter.
BUYER_COUNTY	County of entity receiving shipments from reporter.
TRANSACTION_CODE	"Code determining whether a transaction increases or decreases the reporter's inventory. Post data contains only those with code ""S"" for sale
DRUG_CODE	A four-digit Controlled Substance Code Number that identifies the shipped, regardless of form (pills, patches or nasal sprays, among others) or size (5 milligram, 30 milligram or 80 milligram, among others).
NDC_NO	National Drug Code. The code contains information on the drug product’s manufacturer or distributor, its active ingredient, strength and package size.
DRUG_NAME	Name of drug corresponding with DRUG_CODE. Post data contains only oxycodone and hydrocodone, though full data set contains drugs includingcodeine, morphine and fentanyl, among others.
QUANTITY	Number of packages, weight or volume of shipment. This can take many forms including boxes of boxes or bottles of pills among others.
UNIT	Unit of measurement for QUANTITY. Values include 1: Micrograms, 2: Milligrams, 3: Grams, 4: Kilograms, 5: Milliliters, 6: Liters, D: Dozens, K: Thousands.

```
|ACTION_INDICATOR     |Indication of corrected shipments by reporter. Values include A: adjust, D: 
Delete or I: insert(late-reporter shipment).
|
|ORDER_FORM_NO        |Identifies bath of transactions.
|
|CORRECTION_NO        |Identifies a corrected transaction replacing a previously submitted transact
ion that had been rejected.
|
|STRENGTH             |"One of three values: ""(1) the purity of a bulk rawmaterial  (2) the fracti
onal portion of a standard NDC package size or (3) the percentage by which a package exceeds a stan
dard NDC package size."""                           |
|TRANSACTION_DATE     |Date shipment occurred.
|
|CALC_BASE_WT_IN_GM   |DEA added field indicating the total active weight of the drug in the transa
ction, in grams.
|
|DOSAGE_UNIT          |DEA calculated field indicating number of pills, patches or lozenges, among 
others, shipped as part of the transaction.
|
|TRANSACTION_ID       |Unique record of transaction.
|
|Product_Name         |Trade name of NDC_NO.
|
|Ingredient_Name      |Name of the active ingredient in the drug shipped.
|
|Measure              |Dosage form.
|
|MME_Conversion_Factor|Morphine Milligram Equivalent, or how the specific drug compares to a morphi
ne equivilent.
|
|Combined_Labeler_Name|Cleaned and combined name of entity that manufactured, distributed or relabe
led the drug product in the transaction.
|
|Revised_Company_Name |Cleaned and combined version of Combined_Labeler_Name
|
|Reporter_family      |Cleaned and combined version of REPORTER_NAME.
|
|dos_str              |Strength of dose in milligrams.
|
+---------------------+-----------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
----------------------------------------------------------+
```

Ingest main CSV

In [37]:
```
df = spark.read.option("sep", "\t").option("header", "true").option("inferSchema", "true").csv("g
s://119-f19-opioidbucket/arcos_all_washpost.tsv")
# this take a few minutes.
df.printSchema()
## ideas for speeding up:
#co-locate compute and the buckets
#more nodes; specialize?
```

```
root
 |-- REPORTER_DEA_NO: string (nullable = true)
 |-- REPORTER_BUS_ACT: string (nullable = true)
 |-- REPORTER_NAME: string (nullable = true)
 |-- REPORTER_ADDL_CO_INFO: string (nullable = true)
 |-- REPORTER_ADDRESS1: string (nullable = true)
 |-- REPORTER_ADDRESS2: string (nullable = true)
 |-- REPORTER_CITY: string (nullable = true)
 |-- REPORTER_STATE: string (nullable = true)
 |-- REPORTER_ZIP: integer (nullable = true)
 |-- REPORTER_COUNTY: string (nullable = true)
 |-- BUYER_DEA_NO: string (nullable = true)
 |-- BUYER_BUS_ACT: string (nullable = true)
 |-- BUYER_NAME: string (nullable = true)
 |-- BUYER_ADDL_CO_INFO: string (nullable = true)
 |-- BUYER_ADDRESS1: string (nullable = true)
 |-- BUYER_ADDRESS2: string (nullable = true)
 |-- BUYER_CITY: string (nullable = true)
 |-- BUYER_STATE: string (nullable = true)
 |-- BUYER_ZIP: integer (nullable = true)
 |-- BUYER_COUNTY: string (nullable = true)
 |-- TRANSACTION_CODE: string (nullable = true)
 |-- DRUG_CODE: integer (nullable = true)
 |-- NDC_NO: string (nullable = true)
 |-- DRUG_NAME: string (nullable = true)
 |-- QUANTITY: double (nullable = true)
 |-- UNIT: string (nullable = true)
 |-- ACTION_INDICATOR: string (nullable = true)
 |-- ORDER_FORM_NO: string (nullable = true)
 |-- CORRECTION_NO: string (nullable = true)
 |-- STRENGTH: string (nullable = true)
 |-- TRANSACTION_DATE: integer (nullable = true)
 |-- CALC_BASE_WT_IN_GM: double (nullable = true)
 |-- DOSAGE_UNIT: string (nullable = true)
 |-- TRANSACTION_ID: long (nullable = true)
 |-- Product_Name: string (nullable = true)
 |-- Ingredient_Name: string (nullable = true)
 |-- Measure: string (nullable = true)
 |-- MME_Conversion_Factor: double (nullable = true)
 |-- Combined_Labeler_Name: string (nullable = true)
 |-- Revised_Company_Name: string (nullable = true)
 |-- Reporter_family: string (nullable = true)
 |-- dos_str: string (nullable = true)
```

In [38]:
```
# df.count() # => 178,598,026 records!
```

Paring down to State of interest

Choose your state in the next cell

Note that this is a lazy evaluation.

In [39]:
```
#parameterized! Choose your state here.
_state = 'NH'
df1 = df.filter(df.BUYER_STATE == _state)
# df1.count() #757944. This took a WHILE! Let's save this collect step for later; take our word for
it.
```

In [40]:
```
# How does the data look?
df1 = df1.cache()
df1.head()
```

Out[40]:
```
Row(REPORTER_DEA_NO=u'PB0020139', REPORTER_BUS_ACT=u'DISTRIBUTOR', REPORTER_NAME=u'BURLINGTON DRUG
COMPANY', REPORTER_ADDL_CO_INFO=u'null', REPORTER_ADDRESS1=u'91 CATAMOUNT DR', REPORTER_ADDRESS2=
u'null', REPORTER_CITY=u'MILTON', REPORTER_STATE=u'VT', REPORTER_ZIP=5468, REPORTER_COUNTY=u'CHITTE
NDEN', BUYER_DEA_NO=u'AB3017212', BUYER_BUS_ACT=u'RETAIL PHARMACY', BUYER_NAME=u'BANNON PHARMACY IN
C', BUYER_ADDL_CO_INFO=u'null', BUYER_ADDRESS1=u'109 PLEASANT ST', BUYER_ADDRESS2=u'null', BUYER_CI
TY=u'CLAREMONT', BUYER_STATE=u'NH', BUYER_ZIP=3743, BUYER_COUNTY=u'SULLIVAN', TRANSACTION_CODE=
u'S', DRUG_CODE=9193, NDC_NO=u'53746011805', DRUG_NAME=u'HYDROCODONE', QUANTITY=1.0, UNIT=u'null',
ACTION_INDICATOR=u'null', ORDER_FORM_NO=u'null', CORRECTION_NO=u'null', STRENGTH=u'null', TRANSACTI
ON_DATE=9082008, CALC_BASE_WT_IN_GM=2.27025, DOSAGE_UNIT=u'500.0', TRANSACTION_ID=803008893, Produc
t_Name=u'HYDROCODONE.BITARTRATE 7.5MG/APAP 75', Ingredient_Name=u'HYDROCODONE BITARTRATE HEMIPENTAH
YDRATE', Measure=u'TAB', MME_Conversion_Factor=1.0, Combined_Labeler_Name=u'Amneal Pharmaceuticals
LLC', Revised_Company_Name=u'Amneal Pharmaceuticals, Inc.', Reporter_family=u'Burlington Drug Compa
ny', dos_str=u'7.5')
```

Question: which county experienced the greatest volume change in pills?

Using PySpark to recreate, verify, and expand on the WaPo Investigative team's API work.

In [44]:
```
# ingest population data:
popdata = spark.read.option("sep", ",").option("header","true").option("inferSchema", "true").csv(
"gs://119-f19-opioidbucket/pop_counties_20062012.csv")
# popdata.printSchema()
# popdata.head()
popdata1 = popdata.withColumn("population", popdata.population.cast('int'))
# popdata1.head()
```

In [45]:
```
### lets get sql with it###
## Presentation: expand on why this is valuable.
from pyspark.sql import SQLContext

sqlContext = SQLContext(sc)

df1.createOrReplaceTempView("StateOpioids")

popdata1.createOrReplaceTempView("popdata")
# test = sqlContext.sql("SELECT * FROM popdata")
# test.show(20,False)
```

```
In [46]: county_diffs = sqlContext.sql("with cte1 AS ( \
                                SELECT so.BUYER_COUNTY,\
                                  so.BUYER_STATE,\
                                  SUM(DOSAGE_UNIT) AS TOTAL_PILLS, \
                                  RIGHT(so.TRANSACTION_DATE,4) AS YEAR\
                                  FROM StateOpioids so\
                                  WHERE 1=1 \
                                  GROUP BY so.BUYER_COUNTY, so.BUYER_STATE, YEAR)\
                                SELECT  \
                                  cte1.BUYER_COUNTY,\
                                  MIN(TOTAL_PILLS/population) AS MIN_PER_CAPITA,\
                                  MAX(TOTAL_PILLS/population) AS MAX_PER_CAPITA\
                                FROM cte1\
                                LEFT JOIN popdata pop ON upper(pop.BUYER_COUNTY) = upper(cte1.BUYER_
COUNTY) \
                                                     and CAST(pop.year as int) = CAST(cte1.YEAR
 as int)\
                                                     and upper(pop.BUYER_STATE) = upper(cte1.BUY
ER_STATE)\
                                GROUP BY cte1.BUYER_COUNTY")

# county_diffs.show(20,False)
pDF2 = county_diffs.toPandas()
```

Reproducing results:

```
In [47]: pDF2["MAXDIFF"] = pDF2["MAX_PER_CAPITA"] - pDF2["MIN_PER_CAPITA"]
print(pDF2.sort_values(by=['MAXDIFF'], ascending=False))
```

```
   BUYER_COUNTY  MIN_PER_CAPITA  MAX_PER_CAPITA    MAXDIFF
0          COOS       25.519016       42.175712  16.656696
2       CARROLL       24.620105       40.703084  16.082979
6       GRAFTON       29.983722       43.882564  13.898842
3     STRAFFORD       28.062278       41.103297  13.041019
8       BELKNAP       25.655436       38.280403  12.624967
5    ROCKINGHAM       24.090553       34.604759  10.514206
4      CHESHIRE       20.036571       30.473733  10.437162
1     MERRIMACK       29.474862       39.737589  10.262727
9  HILLSBOROUGH       19.864957       27.545441   7.680484
7      SULLIVAN       23.403079       28.395483   4.992404
```

Answer: Coos County, NH

We have demonstrated that we can compute the maximum change over that seven year period by county very quickly from (mostly) raw data.

Compare against the overall values:

In [48]:

```
state_diff = sqlContext.sql("with cte1 AS ( \
                                SELECT so.BUYER_STATE,\
                                  SUM(DOSAGE_UNIT) AS TOTAL_PILLS, \
                                  RIGHT(so.TRANSACTION_DATE,4) AS YEAR\
                                  FROM StateOpioids so\
                                  WHERE 1=1 \
                                  GROUP BY so.BUYER_STATE, YEAR)\
                              , cte2 AS (\
                                  SELECT SUM(population) AS POPSUM \
                                        , YEAR\
                                        , BUYER_STATE\
                                     FROM popdata\
                                     GROUP BY YEAR, BUYER_STATE)\
                                SELECT  \
                                  cte1.BUYER_STATE,\
                                  MIN(TOTAL_PILLS/POPSUM) AS MIN_PER_CAPITA,\
                                  MAX(TOTAL_PILLS/POPSUM) AS MAX_PER_CAPITA\
                                FROM cte1\
                                LEFT JOIN cte2 pop ON CAST(pop.year as int) = CAST(cte1.YEAR as int)
\
                                                         and upper(pop.BUYER_STATE) = upper(cte1.BUY
ER_STATE)\
                                GROUP BY cte1.BUYER_STATE")

# state_diff.show(5,False)
pDF2A = state_diff.toPandas()
pDF2A["MAXDIFF"] = pDF2A["MAX_PER_CAPITA"] - pDF2A["MIN_PER_CAPITA"]

pDF2A.head()
```

Out[48]:

	BUYER_STATE	MIN_PER_CAPITA	MAX_PER_CAPITA	MAXDIFF
0	NH	24.03082	34.205114	10.174295

Results:

Coos County prescription opioid volume increased at a rate ~60% greater than statewide!

We have shown a way to quickly compute the county and overall state per capita changes in pill volume 2006-2012 (7 years). Researchers or journalists could use this approach to look deeper beyond the WaPo Investigative team's API.

As our out-of-Spark Social Vulnerability Index analysis will show, this is a public health crisis that intersects with other social factors. Coos County's struggles with the opioid crisis and other public health issues likely were exacerbated and potentially contributed to the relative flood of prescription opioids, in some way.

Digging in: Coos County

Potential tool: change in strength and volume by pharmacy over time?

We want to see if the strength and volume by pharmacy over time shows any interesting results: in COOS COUNTY.

Note: State has already been specified in the SQL view.

In [49]:
```
#specifically, we want to look at Coos County, from our initial analysis:
results = sqlContext.sql("SELECT BUYER_NAME, BUYER_ADDL_CO_INFO, BUYER_CITY,\
                         COUNT(TRANSACTION_id) AS REPORT_COUNT,\
                         SUM(DOSAGE_UNIT) AS PILL_SUM \
                         FROM StateOpioids \
                         WHERE 1=1 \
                         AND BUYER_COUNTY = 'COOS'\
                         GROUP BY BUYER_NAME,BUYER_ADDL_CO_INFO, BUYER_CITY") #instantaneous
# results.printSchema()
```

In [50]:
```
# this collect step may take a few minutes as well
# this took ~8 minutes with following settings: central, 1000GB master, 4 500GB helper nodes
results.sort(results.PILL_SUM.desc()).show(20,False)
```

```
+--------------------------------+--------------------------------+----------+------------+---------+
|BUYER_NAME                      |BUYER_ADDL_CO_INFO              |BUYER_CITY|REPORT_COUNT|PILL_SUM |
+--------------------------------+--------------------------------+----------+------------+---------+
|RITE AID OF NEW HAMPSHIRE, INC. |RITE AID #4138                  |COLEBROOK |3831        |2383380.0|
|RITE AID OF NEW HAMPSHIRE, INC. |RITE AID #4127                  |LANCASTER |3246        |2356640.0|
|WAL-MART PHARMACY 10-2634       |null                            |GORHAM    |5323        |1555000.0|
|MAXI DRUG NORTH, INC.           |RITE AID #10287                 |BERLIN    |2492        |997700.0 |
|LAPERLE'S IGA PHARMACY          |null                            |COLEBROOK |1936        |394600.0 |
|RITE AID OF NEW HAMPSHIRE INC   |RITE AID PHARMACY #4157         |GORHAM    |858         |202600.0 |
|RITE AID OF NEW HAMPSHIRE INC   |null                            |BERLIN    |410         |199700.0 |
|PHARMACY OPERATIONS, INC.       |D/B/A THE MEDICINE SHOPPE #1926 |BERLIN    |338         |130300.0 |
+--------------------------------+--------------------------------+----------+------------+---------+
```

Results:

Drilling down - pharmacy detail in Coos County, NH

The raw data shows there are 9 pharmacies in the dataset for Coos County that purchased opioids between 2006 and 2012.

The top hits might be likely targets for pill diversion investigation.

What about trends in dosage strength and pill volume?

In [51]:
```
strength_vals = sqlContext.sql(
                      "SELECT \
                          BUYER_DEA_NO, BUYER_NAME, \
                          RIGHT(TRANSACTION_DATE,4) AS YEAR, \
                          SUM(DOSAGE_UNIT) AS TOTAL_PILLS, \
                          FORMAT_NUMBER(AVG(dos_str),2) AS AVG_DOSE \
                       FROM StateOpioids \
                       WHERE BUYER_COUNTY = 'COOS' \
                       GROUP BY BUYER_DEA_NO, BUYER_NAME, YEAR\
                       ORDER BY BUYER_DEA_NO, YEAR DESC") #instantaneous"

strength_vals.show(50,False)
```

```
+------------+--------------------------------+----+-----------+--------+
|BUYER_DEA_NO|BUYER_NAME                      |YEAR|TOTAL_PILLS|AVG_DOSE|
+------------+--------------------------------+----+-----------+--------+
|BM5180601   |MAXI DRUG NORTH, INC.           |2007|35900.0    |11.91   |
|BM5180601   |MAXI DRUG NORTH, INC.           |2006|81300.0    |12.52   |
|BR3822978   |RITE AID OF NEW HAMPSHIRE INC   |2007|73800.0    |14.56   |
|BR3822978   |RITE AID OF NEW HAMPSHIRE INC   |2006|125900.0   |12.88   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2012|474950.0   |14.61   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2011|494680.0   |13.84   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2010|342250.0   |15.56   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2009|313100.0   |13.32   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2008|259800.0   |11.98   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2007|264300.0   |11.82   |
|BR4157738   |RITE AID OF NEW HAMPSHIRE, INC. |2006|234300.0   |10.79   |
|BR4157788   |RITE AID OF NEW HAMPSHIRE INC   |2009|58600.0    |14.39   |
|BR4157788   |RITE AID OF NEW HAMPSHIRE INC   |2008|56400.0    |14.05   |
|BR4157788   |RITE AID OF NEW HAMPSHIRE INC   |2007|49000.0    |11.77   |
|BR4157788   |RITE AID OF NEW HAMPSHIRE INC   |2006|38600.0    |12.74   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2012|383810.0   |13.32   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2011|381650.0   |12.64   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2010|350580.0   |12.61   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2009|329100.0   |11.86   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2008|318600.0   |12.44   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2007|325400.0   |12.72   |
|BR4157841   |RITE AID OF NEW HAMPSHIRE, INC. |2006|267500.0   |12.53   |
|BR5180601   |MAXI DRUG NORTH, INC.           |2012|212540.0   |13.71   |
|BR5180601   |MAXI DRUG NORTH, INC.           |2011|189160.0   |14.09   |
|BR5180601   |MAXI DRUG NORTH, INC.           |2010|180600.0   |17.14   |
|BR5180601   |MAXI DRUG NORTH, INC.           |2009|141200.0   |13.20   |
|BR5180601   |MAXI DRUG NORTH, INC.           |2008|115700.0   |12.34   |
|BR5180601   |MAXI DRUG NORTH, INC.           |2007|41300.0    |9.50    |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2012|315100.0   |13.68   |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2011|289000.0   |12.48   |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2010|270500.0   |14.83   |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2009|223800.0   |14.14   |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2008|188300.0   |14.78   |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2007|156400.0   |14.99   |
|BW5783623   |WAL-MART PHARMACY 10-2634       |2006|111900.0   |15.03   |
|FL0059887   |LAPERLE'S IGA PHARMACY          |2010|128700.0   |15.01   |
|FL0059887   |LAPERLE'S IGA PHARMACY          |2009|149300.0   |16.06   |
|FL0059887   |LAPERLE'S IGA PHARMACY          |2008|79500.0    |13.80   |
|FL0059887   |LAPERLE'S IGA PHARMACY          |2007|33700.0    |15.00   |
|FL0059887   |LAPERLE'S IGA PHARMACY          |2006|3400.0     |10.62   |
|FP0333942   |PHARMACY OPERATIONS, INC.       |2008|82100.0    |16.12   |
|FP0333942   |PHARMACY OPERATIONS, INC.       |2007|48200.0    |15.10   |
+------------+--------------------------------+----+-----------+--------+
```

Conclusion:

- No clear trends on dosage strength.
- The increase in volume of pills over time in each pharmacy is obvious.

Next:

Deep-dive into who the biggest offenders sourced Opioids from.

There is plenty more to be done here, but we have demonstrated Spark can help with finding the needles in the haystack.

This sort of query could help **journalists or researches determine where to follow up**.

In [52]:
```
### We've identified DEA NO. BR4157738 and BR4157841 as our high offenders.
### Let's look at BR4157738

### Who did they buy from? What were the changes? These would be questions for research.
sellers = sqlContext.sql(
                        "SELECT \
                            BUYER_NAME, BUYER_DEA_NO, REPORTER_DEA_NO, REPORTER_NAME,\
                            RIGHT(TRANSACTION_DATE,4) AS YEAR, \
                            SUM(DOSAGE_UNIT) AS TOTAL_PILLS \
                         FROM StateOpioids \
                         WHERE BUYER_DEA_NO = 'BR4157738' \
                         GROUP BY BUYER_NAME, BUYER_DEA_NO, REPORTER_DEA_NO, REPORTER_NAME, YEAR\
                         ORDER BY YEAR DESC, TOTAL_PILLS DESC")

sellers.show(50,False)
## we see that RITE AID bought most of its pills from McKesson. Could be useful information.
```

```
+--------------------------------+------------+---------------+-------------------------+----+-------
----+
|BUYER_NAME                      |BUYER_DEA_NO|REPORTER_DEA_NO|REPORTER_NAME            |YEAR|TOTAL_P
ILLS|
+--------------------------------+------------+---------------+-------------------------+----+-------
----+
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2012|331540.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RE0356003      |ECKERD CORPORATION       |2012|119510.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RA0287020      |ANDA PHARMACEUTICALS INC|2012|23800.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RA0180733      |ANDA, INC                |2012|100.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2011|384860.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RE0356003      |ECKERD CORPORATION       |2011|109820.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2010|248100.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RE0356003      |ECKERD CORPORATION       |2010|93250.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RA0287020      |ANDA PHARMACEUTICALS INC|2010|900.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2009|227200.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RE0356003      |ECKERD CORPORATION       |2009|46400.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RR0236073      |RITE AID MID-ATLANTIC    |2009|38500.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RA0287020      |ANDA PHARMACEUTICALS INC|2009|1000.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2008|170300.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RR0236073      |RITE AID MID-ATLANTIC    |2008|89500.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2007|180400.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RR0236073      |RITE AID MID-ATLANTIC    |2007|83900.0
|
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |PM0020850      |MCKESSON CORPORATION     |2006|162200.
0   |
|RITE AID OF NEW HAMPSHIRE, INC.|BR4157738   |RR0236073      |RITE AID MID-ATLANTIC    |2006|72100.0
|
+--------------------------------+------------+---------------+-------------------------+----+-------
----+
```

Observations so far:

There is no obvious change over time in the average dose strength of the pills; just in overall volume of pills.

Market changes: It looks like somes pharmacies either stopped operating or selling opioids after '07-'08, which may have driven up the numbers at the remaining large chain locations (Rite Aids, Walmart). This could have been due to rule changes or regulations. Raw volume still increased statewide, despite fewer pharmacy buyers.

The last query shows high-level yearly buying patterns. Unlike Walgreens nation-wide, we see that this Rite Aid bought from regional Rite Aid wholesaler until 2009, when it opted to purchasing solely from major manufacturers/logistics companies, specifically McKesson. **Between 2006 and 2012, McKesson's sales to just this pharmacy DOUBLED.**

What was McKesson's behavior statewide? That could be a next step in investigation.

See more here about WaPo's reporting on Walgreens' practices (https://www.washingtonpost.com/investigations/2019/11/07/height-crisis-walgreens-handled-nearly-one-five-most-addictive-opioids/).

Next Steps:

We want to bring this back down to Earth from Big Data land, for simple stats and geospatial analysis.

Utilize pyspark/sparkSQL to join large pharmacy and opioid datasets

Note that the pharmacy datasets won't be cut down before SparkSQL gets to them - performance is still very good.

In [53]:
```
# ingest Arcos pharmacy national dataset - latlon level
dfLatLon = spark.read.option("sep", ",").option("header", "true").option("inferSchema", "true").csv
("gs://119-f19-opioidbucket/pharmacies_latlon.csv")
# dfLatLon.printSchema()

dfLatLon.createOrReplaceTempView("PharmLatLon")

# ingest Arcos pharmacy national dataset - tract-level
dfTract = spark.read.option("sep", ",").option("header", "true").option("inferSchema", "true").csv(
"gs://119-f19-opioidbucket/pharmacies_tracts.csv")
# dfTract.printSchema()

dfTract.createOrReplaceTempView("PharmTract")
```

In [56]:
```
## next step: write query to aggregate X at pharmacy level, join in tract and latlon data on DEA I
D.
#specifically, we want to look at Coos County, from our initial analysis:
coos_results = sqlContext.sql("SELECT so.BUYER_DEA_NO, pt.GEOID, ll.lat, ll.lon,\
                         SUM(DOSAGE_UNIT) AS PILL_SUM \
                         FROM StateOpioids so\
                         LEFT JOIN PharmTract pt on pt.BUYER_DEA_NO = so.BUYER_DEA_NO \
                         LEFT JOIN PharmLatLon ll on ll.BUYER_DEA_NO = so.BUYER_DEA_NO \
                         WHERE 1=1 \
                         AND so.BUYER_COUNTY = 'COOS'\
                         GROUP BY so.BUYER_DEA_NO, pt.GEOID, pt.TRACTCE, pt.LSAD, ll.lat, ll.lon")

# coos_results.printSchema()
coos_results.show(20,False)
```

```
+------------+-----------+----------+-----------+---------+
|BUYER_DEA_NO|GEOID      |lat       |lon        |PILL_SUM |
+------------+-----------+----------+-----------+---------+
|BR4157841   |33007950500|44.4942167|-71.5720511|2356640.0|
|BR5180601   |33007950800|44.4704933|-71.1805282|880500.0 |
|BW5783623   |33007950900|44.426803 |-71.1941116|1555000.0|
|BM5180601   |33007950800|44.4704933|-71.1805282|117200.0 |
|FP0333942   |33007950800|44.4698775|-71.1809161|130300.0 |
|BR3822978   |33007950800|44.4698775|-71.1809161|199700.0 |
|BR4157738   |33007950200|44.8925853|-71.4982029|2383380.0|
|BR4157788   |33007950900|44.3901658|-71.1830094|202600.0 |
|FL0059887   |33007950200|44.9139077|-71.4927731|394600.0 |
+------------+-----------+----------+-----------+---------+
```

In [57]:
```
state_results = sqlContext.sql("SELECT so.BUYER_DEA_NO, pt.GEOID, ll.lat, ll.lon,\
                         SUM(DOSAGE_UNIT) AS PILL_SUM \
                         FROM StateOpioids so\
                         INNER JOIN PharmTract pt on pt.BUYER_DEA_NO = so.BUYER_DEA_NO \
                         LEFT JOIN PharmLatLon ll on ll.BUYER_DEA_NO = so.BUYER_DEA_NO \
                         WHERE 1=1 \
                         GROUP BY so.BUYER_DEA_NO, pt.GEOID, pt.TRACTCE, ll.lat, ll.lon")
##NOTE that some pharmacies lack tract record; these tended to be very small pharmacies/included sa
les of <8k pills over 7 years.
state_results.show(5,False)
state_results.count()
```

```
+------------+-----------+----------+-----------+---------+
|BUYER_DEA_NO|GEOID      |lat       |lon        |PILL_SUM |
+------------+-----------+----------+-----------+---------+
|AT1545687   |33013032200|43.1984959|-71.532265 |232430.0 |
|FW1834185   |33013038500|43.2806361|-71.8155952|101310.0 |
|AM0439910   |33013032100|43.1941847|-71.5404259|453730.0 |
|BH2504810   |33011012200|42.7751477|-71.4434548|1566360.0|
|BR1058937   |33011011102|42.7139059|-71.4418565|1368420.0|
+------------+-----------+----------+-----------+---------+
only showing top 5 rows
```

Out[57]: 354

Write CSV to bucket for separate (geospatial, etc.) analysis:

In [58]:
```
# type(state_results)
pDF = state_results.toPandas().to_csv("pharmacyAgg.csv", encoding='utf-8', index=False)
```

and now... How powerful is Spark?

BONUS!

This can be altered to look at the national results! With minimal code change:

In [177]:
```
df.createOrReplaceTempView("NationalOpioids")

nat_county_diffs = sqlContext.sql("with cte1 AS ( \
                                SELECT so.BUYER_COUNTY,\
                                  so.BUYER_STATE,\
                                  SUM(DOSAGE_UNIT) AS TOTAL_PILLS, \
                                  RIGHT(so.TRANSACTION_DATE,4) AS YEAR\
                                  FROM NationalOpioids so\
                                  WHERE 1=1 \
                                  GROUP BY so.BUYER_COUNTY, so.BUYER_STATE, YEAR)\
                                SELECT  \
                                  cte1.BUYER_COUNTY,\
                                  cte1.BUYER_STATE,\
                                  MIN(TOTAL_PILLS/population) AS MIN_PER_CAPITA,\
                                  MAX(TOTAL_PILLS/population) AS MAX_PER_CAPITA\
                                FROM cte1\
                                LEFT JOIN popdata pop ON upper(pop.BUYER_COUNTY) = upper(cte1.BUYER_
COUNTY) \
                                                         and CAST(pop.year as int) = CAST(cte1.YEAR
 as int)\
                                                         and upper(pop.BUYER_STATE) = upper(cte1.BUY
ER_STATE)\
                                GROUP BY cte1.BUYER_COUNTY, cte1.BUYER_STATE")

pDF3 = nat_county_diffs.toPandas() #collect step
```

In [179]:

```
pDF3["MAXDIFF"] = pDF3["MAX_PER_CAPITA"] - pDF3["MIN_PER_CAPITA"]
print(pDF3.sort_values(by=['MAXDIFF'], ascending=False))
```

	BUYER_COUNTY	BUYER_STATE	MIN_PER_CAPITA	MAX_PER_CAPITA	\
3110	LEAVENWORTH	KS	68.473825	501.605074	
1066	CHARLESTON	SC	83.294254	381.957373	
2439	MINGO	WV	104.245686	364.808194	
1277	KIMBALL	NE	20.425308	204.717711	
142	FLOYD	KY	90.967226	246.946173	
322	TROUSDALE	TN	49.220555	170.108350	
174	NORTON CITY	VA	238.666667	347.527071	
1978	MARTINSVILLE CITY	VA	191.296128	296.840513	
1831	GALAX CITY	VA	95.478994	191.777554	
71	BACON	GA	57.549148	153.588294	
3119	PICKETT	TN	39.822815	133.967413	
2071	PERRY	KY	123.557876	215.552190	
1771	OWSLEY	KY	64.979970	156.717045	
861	RUTHERFORD	TN	27.817978	117.634200	
969	MARION	AL	66.445629	149.125355	
2419	CLARK	KS	22.362345	104.909008	
916	LESLIE	KY	74.119567	155.173175	
1012	WOLFE	KY	55.195700	135.563168	
256	GRUNDY	TN	99.749876	174.628754	
2415	MORTON	KS	49.050228	123.497522	
2513	DECATUR	TN	104.270770	177.505967	
714	IZARD	AR	48.947950	118.216486	
1826	ESTILL	KY	32.285946	101.350690	
2973	FENTRESS	TN	88.624529	157.639084	
2883	BROOKS	GA	31.673309	99.243745	
61	DEWEY	OK	15.559403	83.118325	
2762	SEQUATCHIE	TN	69.297546	136.096098	
3067	HOPKINS	KY	60.261583	126.125189	
282	ORANGE	TX	53.758414	119.518327	
2694	LEWIS	ID	57.704741	122.420115	
...	...	...	...	...	
1975	CAROLINA	PR	NaN	NaN	
1987	COROZAL	PR	NaN	NaN	
2093	AIBONITO	PR	NaN	NaN	
2110	DORADO	PR	NaN	NaN	
2119	CIDRA	PR	NaN	NaN	
2178	SANTA ISABEL	PR	NaN	NaN	
2280	TOA ALTA	PR	NaN	NaN	
2331	CABO ROJO	PR	NaN	NaN	
2349	MANATI	PR	NaN	NaN	
2383	CIALES	PR	NaN	NaN	
2423	CAGUAS	PR	NaN	NaN	
2431	FAJARDO	PR	NaN	NaN	
2509	null	OH	NaN	NaN	
2545	null	GA	NaN	NaN	
2568	GUAM	GU	NaN	NaN	
2599	SAN GERMAN	PR	NaN	NaN	
2605	null	PR	NaN	NaN	
2701	GUAYANILLA	PR	NaN	NaN	
2744	MAUNABO	PR	NaN	NaN	
2803	null	CA	NaN	NaN	
2819	FLORIDA	PR	NaN	NaN	
2900	SABANA GRANDE	PR	NaN	NaN	
2927	null	FL	NaN	NaN	
2940	GUAYNABO	PR	NaN	NaN	
2996	HATILLO	PR	NaN	NaN	
3000	YAUCO	PR	NaN	NaN	
3076	null	MA	NaN	NaN	
3099	LAS PIEDRAS	PR	NaN	NaN	
3111	SAINT THOMAS	VI	NaN	NaN	
3126	UTUADO	PR	NaN	NaN	

	MAXDIFF
3110	433.131249
1066	298.663119
2439	260.562507
1277	184.292402

```
142   155.978946
322   120.887795
174   108.860405
1978  105.544385
1831   96.298560
71     96.039146
3119   94.144598
2071   91.994314
1771   91.737074
861    89.816221
969    82.679726
2419   82.546664
916    81.053608
1012   80.367468
256    74.878879
2415   74.447293
2513   73.235196
714    69.268536
1826   69.064744
2973   69.014555
2883   67.570435
61     67.558922
2762   66.798553
3067   65.863606
282    65.759913
2694   64.715374
...          ...
1975         NaN
1987         NaN
2093         NaN
2110         NaN
2119         NaN
2178         NaN
2280         NaN
2331         NaN
2349         NaN
2383         NaN
2423         NaN
2431         NaN
2509         NaN
2545         NaN
2568         NaN
2599         NaN
2605         NaN
2701         NaN
2744         NaN
2803         NaN
2819         NaN
2900         NaN
2927         NaN
2940         NaN
2996         NaN
3000         NaN
3076         NaN
3099         NaN
3111         NaN
3126         NaN

[3130 rows x 5 columns]
```

This is why it's important to dig deeper - looks like the Leavenworth example is due to an anomaly in how the shipping data is calculated. (https://www.kcur.org/post/leavenworth-county-kansas-may-not-be-catastrophic-opioid-hotspot-new-data-appear-show#stream/0)

Other observations

There are clearly some anomalies here, but many of these suspect counties are in Appalachia. NH was comparatively less flooded with prescription meds.

/end bonus round!

On to the SVI exploration and remainder of the presentation.